

Q1.

An event-driven, object-oriented programming language lets the programmer create a Graphical User Interface (GUI) from components such as forms and buttons. The components of the GUI are implemented using a class hierarchy and inheritance.

(a) Explain what is meant by inheritance.

.....
.....

(1)

(b) One GUI component is a **Selector**. Selectors come in two different types: **ComboBox** and **ListBox**.

Selector Type	Description
ComboBox	A combo box lets the user make an input either by typing into the box or by picking a single item from a list.
ListBox	A list box lets the user select options from a list. The user cannot type into a list box. There are two different types of list box: <ul style="list-style-type: none"><li data-bbox="576 1178 1342 1279">• SingleSelectionListBox: The user can only select one item from a list. Whenever an item is selected, the previously selected item is deselected.<li data-bbox="576 1312 1366 1413">• MultipleSelectionListBox: The user can select one or more items from a list. Whenever an item is selected, it is added to the list of selected items.

Draw an inheritance diagram for the classes: Selector, ComboBox, ListBox, SingleSelectionListBox and MultipleSelectionListBox.

(3)

- (c) The **Selector** class has data fields **Items** and **NumberOfItemsInList**:
- **Items**: an array that stores the list of strings that will appear in the selector.
 - **NumberOfItemsInList**: a number that indicates how many items there are in the selector.

It also has a procedure that the programmer can call to add an item to the list of strings (**AddItemToList**) and a procedure that is called by the operating system whenever the user selects an item from the list (**SelectItemFromList**).

The **Selector** class does not include a procedure to display the items in the list as the way items are displayed is different for each type of selector.

The class definition for **Selector** is:

```
Selector = Class
  Public
    Procedure AddItemToList
    Procedure SelectItemFromList
  Private
    Items: Array of String
    NumberOfItemsInList: Integer
End
```

A class is to be created for the **ComboBox** type of selector.

The **ComboBox** class needs the following additional data fields:

- **TextTyped**: Stores the characters that have been typed by the user if they have made their input by typing rather than picking an option from the list.
- **SelectedItemNumber**: Stores the position in the list of the item that has been selected by the user, if one has been selected.
- **AllowNonListInputs**: A True or False value that indicates whether the user should be allowed to type in text that is not one of the items in the list.

The class will need to implement the operation of selecting an item from the list differently from the way the **Selector** class implements this operation, but the operation of adding an item to the list will be implemented in the same way by both of these classes.

The class must provide subroutines to:

- display the combo box
- respond to the operating system's notification of a key press
- return the text that has been typed in
- return the selected item number
- set the value of **AllowNonListInputs** flag to True or False, to indicate whether or not the user is allowed to type text that is not in the list.

Write the class definition for the **ComboBox** class.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(5)
(Total 9 marks)

Q2.

An object-oriented program is being written to store details of the hardware devices that are connected to a computer network in a college. This will be used by the network manager to perform an audit of the equipment that the college owns.

Two different types of devices are connected to the network. They are printers and computers. The computers are categorised as being laptops, desktops or servers.

A class **Device** has been created and two subclasses, **Printer** and **Computer** are to be developed. The **Computer** class will have three subclasses: **Laptop**, **Desktop** and **Server**.

(a) Draw an inheritance diagram for the six classes.

(3)

(b) The **Device** class has data fields **MACAddress**, **DeviceName** and **Location**.

The class definition for **Device** is:

```
Device = Class
    Public
        Procedure AddDevice
        Function GetMACAddress
        Function GetDeviceName
        Function GetLocation
    Private
        MACAddress: String
        DeviceName: String
        Location: String
    End
```

The **Computer** class has the following additional data fields:

- **ProcessorName**: Stores the name of the company that manufactured the processor.
- **RAMCapacity**: Stores the capacity of the RAM installed in the computer, in gigabytes.
- **HDDCapacity**: Stores the capacity of the Hard Disk Drive installed in the computer, in gigabytes.

Write the class definition for **Computer**.

.....

.....

.....

.....

.....
.....
.....
.....
.....
.....
.....
.....

(4)

- (c) The **Laptop** class has the additional data field **BluetoothInstalled**. This field will indicate whether or not the laptop is fitted with a Bluetooth module.

Write the class definition for **Laptop**.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

(2)

- (d) Explain what Bluetooth is and give an example of a task for which a laptop user might use Bluetooth.

What Bluetooth is:

.....
.....

(2)

Example use:.....

.....

.....

(1)
(Total 12 marks)

M1.

- (a) A class / subclass has / shares / can access properties and methods of the (parent) class it is derived from;
Building a hierarchy of classes with each child class inheriting access to its parent class' methods and properties;
Relationship between two object (types) in which one object (type) is a kind of the other;

MAX 1

A Just one of properties and methods, do not need both.

A Use of the word "inherits" in the response only if the relationship between parent and subclass is stated explicitly otherwise it is **NE**

A The following as alternatives to properties: fields, attributes, characteristics, data.

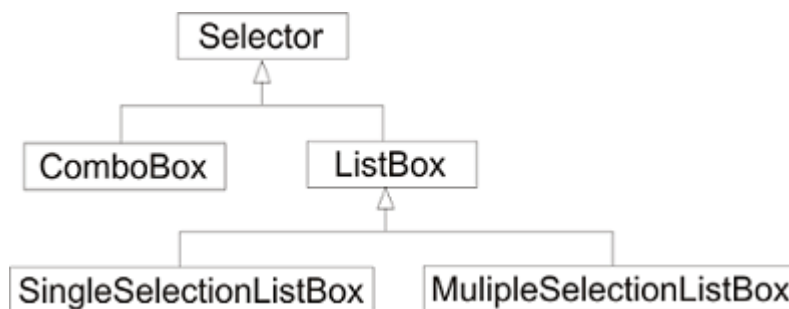
A The following as alternatives to methods: procedures, functions, code.

A The following as alternatives to parent: base, super.

A The following as alternative to child: descendent, subclass, derived.

1

(b)



1 mark for Selector at top of diagram with ComboBox and ListBox directly underneath it and linked to it and no other labels linked to it;

1 mark for ListBox with SingleSelectionListBox and MultipleSelectionListBox directly underneath it and linked to it, and no other labels linked to it (except Selector above);

MAX 1 of the above **2 marks** if any additional links drawn in

1 mark for correctly styled diagram, i.e. lines drawn as arrows and boxes (any shape) around labels; - **This mark is only available if candidate has already achieved at least 1 mark for correct contents of the diagram.**

A arrows drawn as:



A any type of arrowheads.

A diagram rotated through 90 / 180 / 270 degrees.

A arrows draw wrong way round (but cannot get mark for correctly styled diagram).

A class diagrams.

3

```

(c) ComboBox = Class (Selector)
    Public
        Procedure SelectItemFromList
        Procedure Display
        Procedure KeyPressed
        Function GetTextTyped
        Function GetSelectedItemNumber
        Procedure SetAllowNonListInputs
    Private
        TextTyped: String
        SelectedItemNumber: Integer
        AllowNonListInputs: Boolean
End

```

Accept answers that use different notations, so long as meaning is clear.
 Accept any sensible names for subroutines, except
 SelectItemFromList which must have this name as it overrides a procedure in
 the parent class.

1 mark for correct header including name of class (ComboBox) and parent
 class (Selector);
1 mark for overriding the SelectItemFromList procedure (it is not necessary to
 state that overriding is occurring but must be public);
2 marks for defining all 5 other extra functions / procedures needed, all
 identified as being public (keyword public is optional if they are declared
 before variables); **OR 1 mark** if at least 2 of them defined;
1 mark for defining all 3 extra variables, with appropriate data types and
 identified as being private;

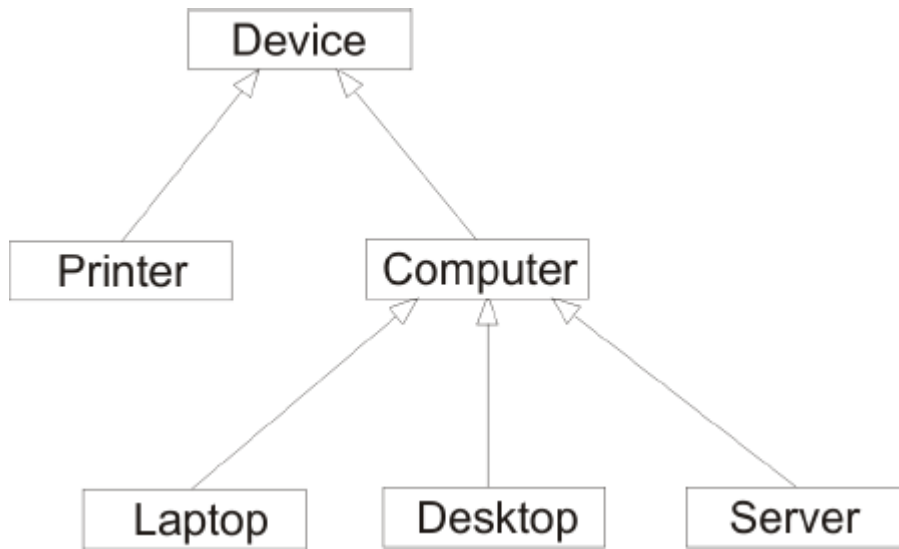
A Array of characters as alternative to string for TextTyped
A Any sensible numeric types for SelectedItemNumber (must be whole
 numbers)
A Answers that indicate separately that each variable is private or each
 method is public
A Two procedures instead of one for setting the value of AllowNonListInputs
 by result, eg, Procedure AllowTextInputs and Procedure OnlyAllowSelection
A Procedure instead of Function and vice-versa
I parameters to methods, minor changes to names that do not affect clarity
R do not award marks for functions / procedures with the same name as
 variables
DPT if any additional functions / procedures / variables declared do not award
 the first of the three marks for correctly defining new functions and variables,
 but award subsequent marks. However, do not penalise answers that include
 any of the following procedures / functions: GetAllowNonListInputs,
 SetTextTyped, SetSelectedItemNumber

5

[9]

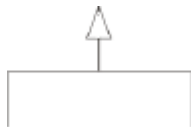
M2.

(a)



1 mark for Device at top of diagram with Printer and Computer directly underneath it and linked to it and no other labels linked to it;
 1 mark for Computer with Laptop, Desktop and Server directly underneath it and linked to it, and no other labels linked to it (except Device above);
 1 mark for correctly styled diagram, i.e. lines drawn as arrows and boxes (any shape) around labels; – **This mark is only available if candidate has already achieved at least one mark for correct contents of the diagram.**

A arrows drawn as:



A filled / empty arrowheads
 A diagram rotated by 90 degrees

3

```

(b)  Computer = Class/Subclass/Extends(Device)           1
      (Public)
      Procedure AddDevice (Override) } 1
      Function GetProcessorName
      Function GetRAMCapacity
      Function GetHDDCapacity } 1
      Private / Protected
      ProcessorName : String
      RAMCapacity : Integer
      HDDCapacity : Integer } 1
      End
  
```

A answers that use different notations, so long as meaning is clear.

1 mark for correct header including name of class and parent class;
 1 mark for redefining the AddDevice procedure;

1 mark* for defining all 3 extra functions needed to read variable values, all identified as being public (keyword public is optional if functions are declared before variables);

1 mark# for defining all 3 extra variables, with appropriate data types and identified as being private;

A any sensible numeric types for RAMCapacity and HDDCapacity, do not have to be whole numbers

A answers that indicate separately that each variable is private or each method is public

R do not award mark for declaring new functions if any of the functions have the same name as the variables

I parameters to methods, minor changes to names that do not affect clarity

* – Do not award this mark if any extra functions / procedures have been declared, except for functions that would set values e.g. SetProcessorName or an incorrectly named procedure to add e.g. AddComputer

– Do not award this mark if any extra variables have been declared

4

(c)

```
Laptop = Class/Subclass (Computer)           1
      (Public)
      Procedure AddDevice (Override)          1
      Function GetBluetoothInstalled
      Private / Protected                    1
      BluetoothInstalled : Boolean
      End
```

1 mark for correct header including name of class and parent class;

Max 1 of the following two marks:

1 mark* for redefining the AddDevice procedure;

1 mark* for:

- defining the GetBluetoothInstalled function needed to read this value, identified as being public (keyword public is optional if function is declared before variable)
- defining the BluetoothInstalled variable with an appropriate data type as being private.

A Boolean or whole number types for BluetoothInstalled but reject string, character or real number types

A Different sensible name for GetBluetoothInstalled function e.g. CheckBluetoothInstalled, IsBluetoothInstalled

A answers that indicate separately that each variable is private or each method is public

I parameters to methods, minor changes to names that do not affect clarity

I addition of any extra functions or variables

* Do not award this mark if any extra functions / procedures / variables declared, except for a SetBluetoothInstalled procedure.

2

- (d) **What** (2 marks):
Wireless/RF (protocol/standard/technology);
For exchanging data over short distances // for creating
Personal Area Network;
NE “uses waves” for “wireless”

Example (1 mark):

Any sensible example, related to the use of Bluetooth with the laptop e.g.
synchronising contacts between phone/ laptop, sending photographs from
phone to laptop, Bluetooth mouse, Bluetooth headset / headphones (used with
laptop) etc;

NE connecting to wireless network

NE mouse

*If the example makes clear that the technology is wireless, but this is not
explicitly stated in the “What” part of the response then the “Wireless” mark
should be awarded in the “What” part.*

3

[12]